
Object Recognition Based on Amounts of Unlabeled Data

FUQIANG LIU

FQLIU@OUTLOOK.COM

Beijing Institute of Technology, Beijing 100081 CHINA

FUKUN BI

BIFUKUN@NCUT.EDU.CN

North China University of Technology, Beijing 100144 CHINA

LIANG CHEN

CHENL@BIT.EDU.CN

Beijing Institute of Technology, Beijing 100081 CHINA

Abstract

This paper proposes a novel semi-supervised method on object recognition. First, based on Boost Picking (Fuqiang Liu, 2016), a universal algorithm, Boost Picking Teaching (BPT), is proposed to train an effective binary-classifier just using a few labeled data and amounts of unlabeled data. Then, an ensemble strategy is detailed to synthesize multiple BPT-trained binary-classifiers to be a high-performance multi-classifier. The rationality of the strategy is also analyzed in theory. Finally, the proposed method is tested on two databases, CIFAR-10 and CIFAR-100 (Krizhevsky, 2009). Using 2% labeled data and 98% unlabeled data, the accuracies of the proposed method on the two data sets are 78.39% and 50.77% respectively.

1. Introduction

High-level computer vision applications like detection and recognition have been developing rapidly. Supervised methods (Mohri et al., 2012) contribute a lot and representative works include DPM (Felzenszwalb et al., 2010), Alexnet (Krizhevsky et al., 2012b), Highway Network (Rupesh Kumar Srivastava, 2015), Residual Network (Kaiming He, 2015) and so on. Large and open data sets like ImageNet (Fei-Fei, 2010) and Microsoft COCO (Lin et al., 2014) greatly contribute to this boom. And crowds (Attenberg et al., 2011; Burton et al., 2012), who labeled these data, also play an important role in the development.

But the size of unlabeled data in practice is much larger than that of labeled data and it is too expensive to label a large number of data by humans, especially the data are generated all the time. Humans do not need so many labeled data like the computer to learn a new knowledge or concept. A baby gradually knows the world from his practices and parent's guide. In this process, unlabeled data might be more important. As a result, this paper researches in how to mainly use unlabeled data to teach computer recognition.

To begin with, this paper presents the theoretical basis (see section 2) of the proposed method. Kalal's works (Kalal et al., 2010; 2012) presented P-N Learning to update object models in tracking application. Kalal proved that P-N experts can train a strong classifier only if eigenvalues of the transformation matrix are all smaller than 1 (equation 2). Liu's work (Fuqiang Liu, 2016) pointed out that P-N Learning ignored the bias of supervised models in training set and proposed Boost Picking to convert supervised classification to semi-supervised classification. Using Boost Picking, two weak classifiers could train a strong classifier mainly by unlabeled data. Liu derived that Boost Picking could work effectively

even taking the bias into account. Furthermore, Liu theoretically proved that Boost Picking would train a supervised classification model mainly by un-labeled data as effectively as the same model trained by all labeled data, only if recalls of the two weak classifiers are both greater than 0 and sum of their precisions is greater than 1 (equation 4).

And then, referring to Boost Picking, we design a method named "Boost Picking Teaching (BPT)" for binary-classification (see section 3). BPT includes a binary-classifier(named BPT-trained binary-classifier) and two "teachers". Like a baby knowing the world, BPT improves the performance of the binary-classifier by practicing with unlabeled data and teachers' guides. Different from Boost Picking, BPT does not need any labeled data to initialize the binary-classifier, which is like that a baby knowing the world originally from unlabeled data. BPT achieves that a strong binary-classifier could be trained totally by unlabeled data under a specific and loose condition. The condition is as same as that in Boost Picking (Fuqiang Liu, 2016) (equation 4). The binary-classifier classifies those unlabeled data and then "teachers" find the errors and correct the classifier by retraining. This step is like bootstrap (Sung & Poggio, 1995), but BPT uses estimated labels rather than true labels. Liu's work theoretically guarantees that even though the "teachers" are not good enough, the binary-classifier could be trained to be a strong one just by repeating the above step. However, in the recognition application (see section 5), we still use a few labeled data to train qualified "teachers". Under the relatively complicated recognition circumstance, it would be a little bit difficult to make sure the "teachers" satisfy the above condition (equation 4), if the "teachers" are designed based on un-supervised methods.

Furthermore, to achieve recognition, this paper proposed a high-performance multi-classifier based on BPT-trained binary-classifiers (see section 4). First, we briefly discuss why it is essential to combine multiple BPT-trained binary-classifiers rather than directly train a multi-classifier like BPT. Then, we details the framework of the ensemble classifier (named BPT-multi-classifier) as well as the synthesizer (ensemble strategy). A novel idea is that the synthesizer is composed by BPT-binary classifiers. It is also theoretically analyzed that the framework of BPT-multi-classifier is reasonable.

Based on BPT-multi-classifier, this paper proposes a practical implementation on object recognition (see section 5). The basic classification model of BPT-trained binary classifier is a support vector machine(SVM) (Suykens & Vandewalle, 1999). And K-means (Coates & Ng, 2012) is adopted to extract features from the images. As for the "teachers" in BPT, they are achieved by logistic regression models (Hosmer Jr & Lemeshow, 2004) that are trained by a few labeled data. And features that "teachers" use are extracted by Histogram of Oriented Gradient (HOG) (Dalal & Triggs, 2005). CIFAR-10 and CIFAR-100 (Krizhevsky, 2009) are involved in the experiment (see section 6) to test the effectiveness of the proposed method. The accuracies of the proposed method on the two databases are 78.39% and 50.77% respectively. In this test, the proposed method use only 2% labeled data and 98% unlabeled data (the size of the two data sets are both 60000).

The BPT-multi-classifier could be regarded as a semi-supervised method because labeled data and unlabeled data are both used. But different from previous object recognition works based on semi-supervised methods (Chen et al., 2013; Rosenberg et al., 2005; Fergus et al., 2009; Prest et al., 2012), BPT-multi-classifier does not involve any web data. And unlike one-shot learning (Li et al., 2006; Socher et al., 2013) and baby learning (Liang et al., 2014), the proposed work does not use any prior knowledge or pre-trained model. In fact, the thought of the proposed method is different from the thoughts of previous semi-supervised methods. The basic units of BPT-multi-classifier are some traditional supervised models (SVM and logistic regression). The key idea of our work is that weak classifiers could train a strong classifier just using unlabeled data and their estimated labels (the weak classifiers may be trained by a few labeled data).

2. Theoretical Basis

This section presents the theoretical basis of BPT.

In (Kalal et al., 2012), new forms of the tracked object are updated by P-N Learning. P-N Learning adopts P-N experts to estimate errors of the detector. P-expert finds the false negatives from the negative outputs of the detector and N-expert finds the false positives from the positive outputs. Kalal theoretically proved that P-N Learning could train a detector whose error converges into zero. Define $\vec{e}(k)$ as the errors in kth iteration,

$$\vec{e}(k) = [\alpha(k), \beta(k)]^T \quad (1)$$

where $\alpha(k)$ represents false positives and $\beta(k)$ represents false negatives in outputs.

Define P^+, R^+ as the precision and the recall of P-expert respectively. Similarly, define P^-, R^- as the precision and the

recall of N-expert. The recursive equations are as equation (2) (Kalal et al., 2012).

$$\begin{cases} \alpha(k+1) = \alpha(k) - R^- \cdot \alpha(k) + \frac{1-P^+}{P^+} R^+ \cdot \beta(k) \\ \beta(k+1) = \beta(k) - R^+ \cdot \beta(k) + \frac{1-P^-}{P^-} R^- \cdot \alpha(k) \end{cases} \quad (2)$$

Reconstruct equation (1) and (2),

$$\begin{aligned} \vec{e}(k+1) &= M \vec{e}(k) \\ M &= \begin{bmatrix} 1-R^- & \frac{1-P^+}{P^+} R^+ \\ \frac{1-P^-}{P^-} R^- & 1-R^+ \end{bmatrix} \end{aligned} \quad (3)$$

Assuming λ_1, λ_2 are eigenvalues of transformation matrix M , Kalal concludes that \vec{e} will converge to zero only if λ_1, λ_2 are all smaller than 1 (Kalal et al., 2010). In practice, the classifier almost cannot be trained to be errorless even through P-N experts could find out all false positives and false negatives, because bias (Geman et al., 1992; James, 2003) always exists.

Supposing x is an example from a feature-space X and y is its corresponding label from a label space $Y = \{1, -1\}$, X_l is a labeled data set with its label set L_l , and X_u is an unlabeled data set. Supposing that L_u are the in-existent true labels of X_u , define $F : x \rightarrow y$ as an supervised learning model trained by 100% labeled data set $X_{(100\%)} = \{X_l, X_u\}$ with $L_{(100\%)} = \{L_l, L_u\}$. Boost Picking (Fuqiang Liu, 2016) aims to train an effective model $f \rightarrow F$ by X_l, L_l and X_u . In Boost Picking (Fuqiang Liu, 2016), two weak classifiers pick out the false positives and false negatives from the outputs of f just like what P-N experts do. Then put these estimated errors into the re-training set with their estimated labels and retrain f like bootstrap (Sung & Poggio, 1995) (bootstrap uses true labels but Boost Picking use the estimated labels). Liu proved that $f \rightarrow F$ in the iteration if equation 4 is satisfied, both in theory and experiment. Boost Picking could still work even considering the bias.

$$\begin{cases} R^+ R^- \neq 0 \\ P^+ + P^- > 1 \end{cases} \quad (4)$$

Equation 4 guides how to use two weak classifiers to train a strong binary classifier. Based on this conclusion and referring to Boost Picking, this paper proposes a modification, Boost Picking Teaching (BPT), for binary classification. And furthermore, BPT is expanded to multiple-classification and a complicated application, object recognition.

Boost Picking is to train a strong binary classifier by labeled data X_l and unlabeled data X_u , while BPT is to train a strong binary classifier by unlabeled data X_u . Unlike Boost Picking, BPT does not need labeled data to initialize the binary classifier. Because the initial set and re-training set of Boost Picking rely on a certain number of labeled data, the classifier trained by Boost Picking would be over-fitting if labeled data are not enough. In fact, Boost Picking needs 25% labeled data as well as 75% unlabeled data to train a strong classifier. But in theory, BPT does not need labeled data. It should be noted that a few labeled data might be used in BPT to train qualified "teachers". Because the test sets in (Fuqiang Liu, 2016) are relatively simple and small, Boost Picking just used an unsupervised method to train the two weak classifiers, FP and FN Finders. But the proposed method is tested on large databases in this paper, and this is why the two weak classifiers, "teachers", in BPT need a few labeled data to train themselves.

3. Boost Picking Teaching for Binary Classification

This section details BPT for binary classification. BPT could train a strong classification f just using unlabeled data X_u in theory. The framework of BPT is showed as Fig 1. Though it refers to the framework of Boost Picking (Fuqiang Liu, 2016), there are some modifications to achieve the goal, training a supervised binary-classifier without labeled data.

Supposing f is the binary classifier parameterized by Θ . The first step of BPT is that Θ are initialized as $\mathbf{0}$. Actually, there is no key difference between zero initialization and random initialization in BPT.

Second, one part of unlabeled data are classified by the binary classifier. And two "teachers", FP Finder and FN Finder, pick out false positives FP and false negatives FN in the outputs. Errors of the two "teachers" are allowed, only if the

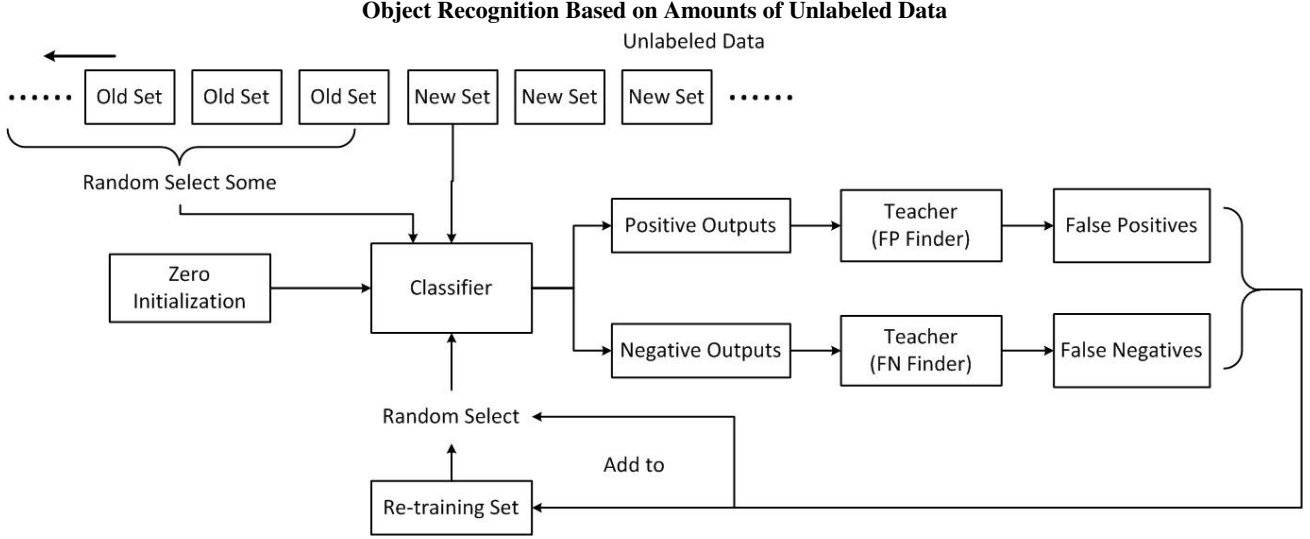


Figure 1. Framework of BPT for Binary Classification

”teachers” satisfy equation 4. Then, these estimated errors with their amended labels (the labels of examples in FP are set as -1 and the labels of examples in FN are set as 1) are put into the re-training set X_R , $X_R = X_R \cup \{FP, FN\}$. And the classifier f is re-trained by X_r , where $X_r \subset X_R$ and $\{FP, FN\} \subset X_r$. Next, go back to the beginning of this step and repeat classifying, correcting and re-training. The iteration process stops when the performance of the classifier is stable. At this time, the classifier is trained to be parameterized as θ_{S_5} . In every generation, only part of examples in X_R are used in re-training because X_R would gradually grow to be a large set in the iteration proceeding. $X_r = \text{randomselect}(X_R) \cup \{FP, FN\}$. Considering efficiency, examples from $\{FP, FN\}$ are all involved in X_r . To avoid being over-fitting, X_r is generated by randomly selecting X_R . The idea that put mis-classified examples into training set resembles bootstrap (Sung & Poggio, 1995). But unlike bootstrap, BPT does not need the true labels of the data.

Third, a new part of unlabeled data are classified by the classifier. The next operations are almost similar with the second step. But there are two special operations. One is that the classifier is parameterized as θ_S initially. Another one is that some examples randomly selected from previous parts are also classified by the binary classifier like Fig 1. Repeat this step until no more data to input. If we do not take the labeled data might used to train ”teachers” into account, BPT could train an effective binary-classifier just using un-labeled data, only if the ”teachers” satisfy equation 4.

Algorithm 1 Boost Picking Teaching (Binary Classification)

<p>Initialization:</p> <p>$\Theta_S = \mathbf{0}$</p> <p>$X_R = \emptyset, X_P = \emptyset$</p> <p>repeat</p> <p> Input: un-labeled data X_{u_i}</p> <p> $\Theta_s = \Theta_S$</p> <p> $X_P = X_P \cup \{X_{u_{i-1}}\}$</p> <p> $Now = f(X_{u_i} \Theta_s)$</p> <p> repeat</p> <p> $Pre = Now$</p>	<p>$V = f(\text{ransel}(X_P) \Theta_s)$</p> <p>$\{FP, FN\} = \text{Teachers}(\{Pre, V\})$</p> <p>$X_R = X_R \cup \{FP, FN\}$</p> <p>$X_r = \text{ransel}(X_R) \cup \{FP, FN\}$</p> <p>$\Theta_s = \text{ReTrain}(f, X_r, \Theta_s)$</p> <p>$Now = f(X_{u_i} \Theta_s)$</p> <p>until $\sum Pre - Now < \text{threshold}$</p> <p>$\Theta_S = \Theta_s$</p> <p>until noInput</p>
--	--

For Boost Picking (Fuqiang Liu, 2016), the training set includes the labeled data in every generation, which causes that the classifier relies on those labeled data too much. BPT adopts random selection to solve this problem but random selection increases the computational complexity of BPT. BPT inputs unlabeled separately, which is convenient to expand training set and involve new data. But BPT is not a critical online learning (Shalev-Shwartz, 2012; Bertsekas, 2015; Brooms, 2006) method because BPT does not abandon any previous data. The pseudocode of BPT is shown in algorithm 1. *ransel* represents ”Random Select”. The convergence condition is that the classifier is stable in the current part of unlabeled data. It is set empirically because the binary classifier would converge more quickly compared to the condition that the classifier should be stable in the whole unlabeled data.

Fig 2 shows the performance of BPT with synthetic ”teachers” whose precisions and recalls are specific. The implementa-

tion of the binary classifier for recognition is declared in section 5. Examples form one class are regarded as positives, the others are regarded as negatives. CIFAR-10 (Krizhevsky, 2009) is used in this ideal experiment. Recalls and Precisions of the synthetic "teachers" are 0.6. We compute the accuracy, recall, precision and $F1 = \frac{2 \cdot P \cdot N}{P+N}$ of the binary-classifier in training set(100% unlabeled data). When the unlabeled data have not yet input, the binary-classifier is parameterized as $\mathbf{0}$. Because the data includes 10 classes and the number of every class is almost equal, the accuracy of the classifier is approximately 0.9 initially. Recall, precision and $F1$ are all 0 at the beginning. According to Fig 2, all these indexes increase with the proceeding of BPT (with synthetic "teachers"). This ideal experiment proved that BPT could train an effective binary-classifier just using un-labeled. When the "teachers" are not easy to trained to satisfy equation 4, a few labeled data might be used to train qualified "teachers". And in this time, BPT becomes a semi-supervised method.

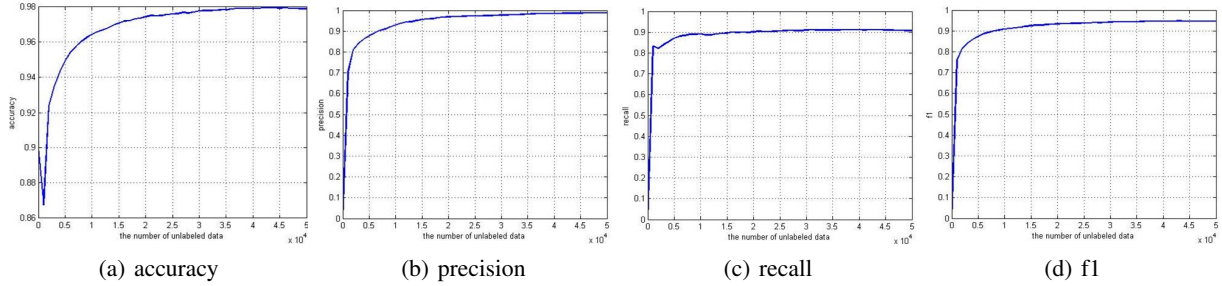


Figure 2. The performance of BPT with synthetic "teachers".

4. Multi-Classification Based on BPT

This section details how to construct a multi-classifier based on the BPT-trained binary-classifiers. First, the reason of converting multi-classifier to multi binary-classifiers is briefly proposed. Second, this part presents the multi binary-classifiers. Third, the ensemble strategy on combining these binary-classifiers to be a multi-classifier is detailed.

4.1. Why Not Directly Train a Multi-Classifier

Suppose that f' is a multi-classifier, x is an example from a feature-space X and y is its corresponding label from a label space $Y = \{1, 2, \dots, N\}$, where N is the number of the class. This part discusses whether f' could be directly guided by "teachers".

Assume that $FalseI$ means the examples which are mis-classified as "I" by f' , where $I \in \{1, 2, \dots, N\}$. Like BPT, there should be a "teacher" to find out the $FalseI$. Suppose that T_i is the "teacher" who finds "i" examples from $\{x|f'(x) \neq i\}$, where $i \in \{1, 2, \dots, N\}$. Then assuming that $TeacherI$ picks out $FalseI$ from $\{x|f'(x) = I\}$, $TeacherI = ensemble\{T_i|i \neq I \text{ and } i \in \{1, 2, \dots, N\}\}$. Assuming P_i is the precision of T_i and R_i is the precision of T_i , resembling equation 2, the recursive equation of the errors of f' are written as equation 5.

$$\left\{ \begin{array}{l} e_1(k+1) = e_1(k) - \sum_{i \neq 1} R_i \cdot a_{i,1} \cdot e_1(k) + \sum_{j \neq 1} \frac{1-P_1}{P_1} R_1 \cdot e_j(k) \\ e_2(k+1) = e_2(k) - \sum_{i \neq 2} R_i \cdot a_{i,2} \cdot e_2(k) + \sum_{j \neq 2} \frac{1-P_2}{P_2} R_2 \cdot e_j(k) \\ \vdots \\ e_N(k+1) = e_N(k) - \sum_{i \neq N} R_i \cdot a_{i,N} \cdot e_N(k) + \sum_{j \neq N} \frac{1-P_N}{P_N} R_N \cdot e_j(k) \end{array} \right. \quad (5)$$

where e_i represents $FalseI(i = I)$. Supposing $l(x)$ represents the true class of x , $a_{i,j}$ represents the proportion between $\{x|f'(x) = j \text{ and } l(x) = i\}$ and $\{x|f'(x) = j\}$, where $i \neq j$. And it is obvious that $\forall j \in \{1, 2, \dots, N\}$, $\sum_{i \neq j} a_{i,j} = 1$.

Assume $\vec{e} = \{e_1, e_2, \dots, e_N\}$. Like equation 2, equation 5 could be reconstructed to be equation 6.

$$\vec{e}(k+1) = M\vec{e}(k) \quad (6)$$

The transformation matrix M is a matrix whose size is $N \times N$. $M(i, j) = 1 - \sum_{i \neq j} R_i \cdot a_{i,j} \cdot e_j(k)$ when $i = j$ and $M(i, j) = \frac{1-P_i}{P_i} R_i \cdot e_j(k)$ when $i \neq j$. Based on the well founded theory of dynamic system (K. Zhou & Glover, 1996), if the absolute values of eigenvalues of M are all less than 1, the error of f' would converge to 0; otherwise the error would diffuse.

Here, simplify M to be easily analyze. Assume that $\forall i, j \in \{1, 2, \dots, N\}$ and $i \neq j$, $a_{i,j} = \frac{1}{N-1}$. This assumption means that the true labels of the examples in $FalseI$ might be any other class in equal probability. Furthermore, suppose that the precision and recall of every T_i are equal, $\forall i$, $P_i = P$, and $R_i = R$, where $P, R \in [0, 1]$. Suppose that λ_i is the eigenvalue of the transformation matrix M . The expressions of λ_i is equation 7.

$$\begin{cases} \lambda_1, \dots, \lambda_{N-1} = \frac{P-R}{P} \\ \lambda_N = \frac{P + (N-1) \cdot R - N \cdot P \cdot R}{P} \end{cases} \quad (7)$$

It is obvious that

$$\begin{aligned} \max(|\lambda|) &< 1, \text{ when } P > \frac{N-1}{N} \\ \max(|\lambda|) &= 1, \text{ when } P = \frac{N-1}{N} \\ \max(|\lambda|) &> 1, \text{ when } P < \frac{N-1}{N} \end{aligned} \quad (8)$$

no matter what R is. It is concluded that if the number of class N increases, P must increase to make sure that $\max(|\lambda|)$ less than 1, which means the recursive equation 6 could converge to 0 in the iteration proceeding.

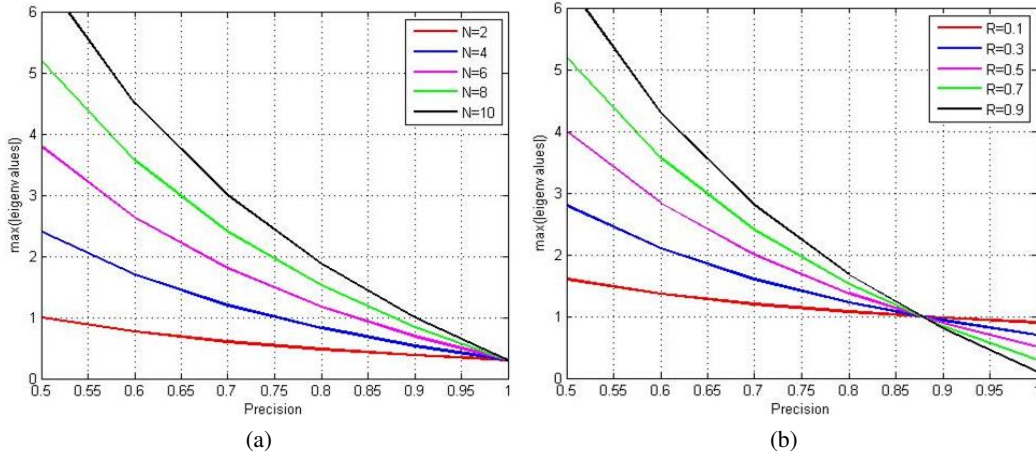


Figure 3. Analysis on the relationship between the maximum absolute value of eigenvalues and (P, R, N) .

To better illustrate the relationship between the maximum absolute value of these eigenvalues and $(Recall, N)$, we simulate the eigenvalues in various conditions and Fig 3 (a-b) are the simulated results. In Fig 3 (a), recall R is set as 0.6 constantly. (a) shows that the precision need to increase to guarantee that errors of the binary-classifier converge to 0, which is as same as the conclusion based on equation 8. In Fig 3 (b), N is set as 8 constantly and R is varied. (b) shows that if the maximum absolute value of eigenvalues is 1, the precision would be certain no matter what R is. This conclusion also matches equation 8.

Based on equation 8 and Fig 3, the condition of "teachers" would be rigorous if training a multi-classifier using unlabeled data like BPT, especially the number of class N is large. Compared to equation 4, the above condition 8 does not mean that the "teachers" are weak classifiers anymore. As a result, directly training a effective qualified multi-classifier using

Table 1. Performance of each binary-classifier trained by BPT with synthetic "teachers" whose precisions and recalls are 0.6

	f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8	f_9	f_{10}
accuracy	0.982	0.988	0.973	0.968	0.977	0.976	0.985	0.980	0.989	0.987
precision	0.974	0.985	0.967	0.956	0.969	0.964	0.978	0.978	0.987	0.984
recall	0.934	0.953	0.885	0.858	0.910	0.909	0.952	0.926	0.959	0.951
$F1$	0.953	0.968	0.924	0.904	0.938	0.936	0.965	0.951	0.973	0.968

unlabeled data like BPT is relatively hard. It is better to combine multiple BPT-trained binary-classifiers to be an effective multi-classifier.

4.2. From Multi Classification to Multi Binary-Classification

The multi-classification problem is separated to multiple $1 \text{ vs } N - 1$ binary-classification problems, and then synthesize results of the multiple binary-classifiers. This part details the multiple binary-classifiers.

Assuming f_1, f_2, \dots, f_N are the binary-classifiers used to compose the multi-classifier, each classifier is trained by BPT using unlabeled data(see section 3). It should be noted that BPT might need a few labeled data to train qualified "teachers", especially when the database is complicated and large. $f_i, i \in \{1, 2, \dots, N\}$ is to find out examples those belong to i class from all data.

There are another ideal experiment to show the performances of these binary-classifiers. As the ideal experiment in section 3, the "teachers" in BPT are synthetic and their precision and recalls are all 0.6. We use CIFAR-10 (Krizhevsky, 2009) as the experimental data and the specific implementation (including feature extraction and supervised learning model) of the binary-classifier is proposed in section 5. Table 1 shows the accuracy, precision, recall and $F1$ of each BPT-trained binary-classifier in training data(100% unlabeled data). Under ideal condition, these binary-classifiers could be trained very well.

4.3. Ensemble Strategy

This section introduces how to synthesize multiple BPT-trained binary-classifiers to be a high-performance multi-classifier. The main reason of ensemble is that there might be more than one binary-classifiers classify a same one example as a positive output, which means one example might be classified to be multiple classes. Fig 4 is the pipeline of the multi-classifier including both multiple PBT-trained binary-classifiers and the ensemble strategy.

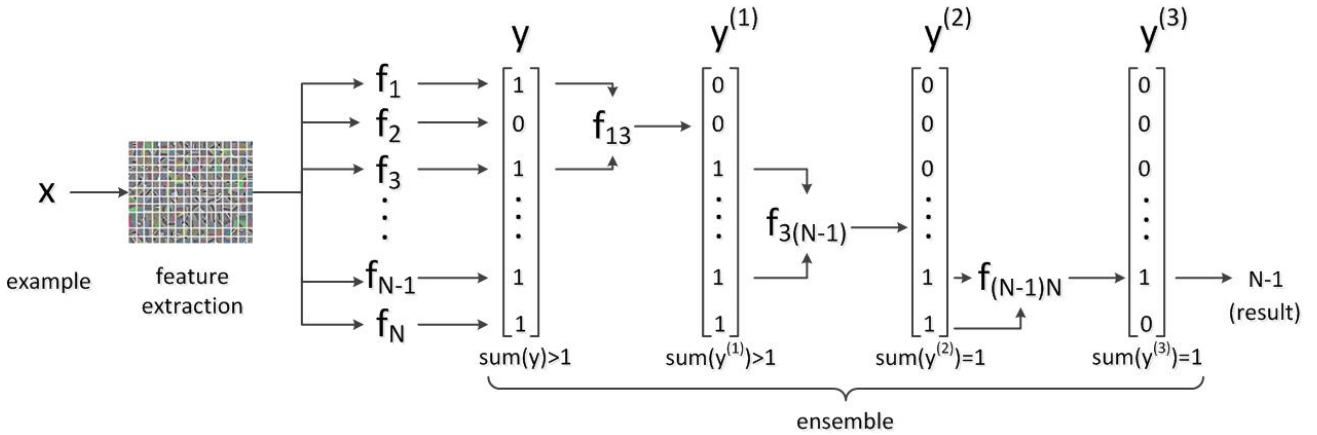


Figure 4. the pipeline of the multi-classifier

Supposing the data contains examples from N classes, N binary-classifiers ($1 \text{ vs } (N - 1)$) are train well by BPT (see section 3). Assuming x is an example from the data set, $y = \{f_1x, f_2x, \dots, f_Nx\}'$ and it is possible that $\text{sum}(y) \geq 1$. After the ensemble module, there should be $\text{sum}(y) = 1$. The ensemble module is composed by $\frac{N(N-1)}{2}$ binary-

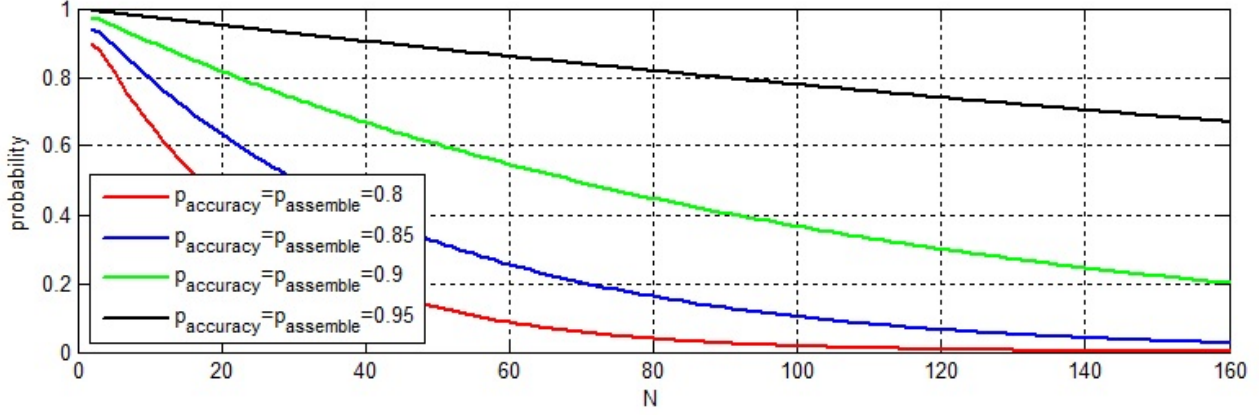


Figure 5. the relationship between the probability of correct classification and N .

classifiers. Supposing f_{ij} is a binary-classifier that is trained by $X_{ij} = x \in X_u | f_i(x) + f_j(x) = 1$ and $i \neq j$ using BPT, f_{ij} is aimed to determine whether x belongs to class i . The example x is input into the N binary-classifiers and the output is a $N \times 1$ vector y . $y(i) = 1$ represents the example is regarded as class i by the classifier f_i , where $i \in \{1, 2, \dots, N\}$. When $\text{sum}(y) > 1$, x is classified as positives by more than one classifiers. Take Fig 4 for instance, after the beginning N classifiers, x is judged by f_{13} firstly to determine whether x belongs to class 1. After then, the number of 1 in $y^{(1)}$ is one less than that in y . Repeat input to other classifier f_{ij} where $y(i) = y(j) = 1$ until $\text{sum}(y) = 1$. Finally, the class of x determined as c , $y(c) = 1$ and $\text{sum}(y) = 1$. If the initial y has no 1, the class of the related x would be assigned randomly.

Here, we discuss the accuracy of this structure. Assuming that the accuracy of f_i , $i \in \{1, 2, \dots, N\}$ is p_{accuracy} , the probability of that one example is correctly classified by the supposed binary-classifier but mis-classified by other K binary-classifiers are shown as equation 9. The output of the ensemble module would be right only under this condition. K could be 0 and it means that the example is correctly classified.

$$p_K = C_{N-1}^K \cdot p_{\text{accuracy}}^{N-K} \cdot (1 - p_{\text{accuracy}})^K \quad (9)$$

Then the example is input to the ensemble module. Assuming the accuracy of each f_{ij} is p_{assemble} equally, only if results of all used classifiers f_{ij} , $f_i(x) \cdot f_j(x) = 1$ in the module are right, the example would be classified correctly. Taking no account of the random assignment, the probability of that the result of the multi-classifier is correct is shown as equation 10.

$$p_{\text{correct}} = \sum_{K=0}^{N-1} p_K \cdot p_{\text{assemble}}^K \quad (10)$$

Fig 5 shows the relationship between the p_{correct} and the number of class N . With the increase of N , the probability of correct classification is decreasing. As a result, this structure would work well when the number of classes are not too many. In contrast, it would perform terribly if this structure is used to classify the examples from a data including too many classes. Usually, if the data contains less than 20 classes, the above ensemble strategy is a satisfactory solution to synthesize these multiple BPT-trained binary-classifiers.

5. Recognition Implementation

This section introduces the specific implementation of the multi-classifier in section 4 for object recognition. It includes three parts: feature extraction, basic classifier and the implementation of "teachers".

The feature extraction is based on K-means (Coates & Ng, 2012), an unsupervised learning method. Support vector machine (SVM) (Suykens & Vandewalle, 1999) is chosen as the basic classifier for BPT-trained binary-classifier. As

for "teachers", we use logistic regression as the classification model, Histogram of Oriented Gradient(HOG) (Dalal & Triggs, 2005) as the feature extraction method, and principal component analysis (PCA) (Jolliffe, 2010) to decrease the dimensionality. Both the classification model and the feature extraction method are different from those of the basic classifier. These differences could improve the diversity.

Some labeled examples are reserved to train these "teachers". Every basic classifier needs two "teachers" to train itself. Furthermore, each classifier in the ensemble module is also need two "teachers". The two "teachers" are composed by one positive "teacher" and one negative "teacher". The positive "teacher" is to pick out false negative examples from the negative outputs of the binary-classifier. The negative "teacher" is to pick out false positive examples from the positive outputs. These "teachers" are trained by some examples randomly selected from these reserved labeled data. And the "teachers" are frequently re-trained in the proceeding of BPT, which aims to make sure that the "teachers" satisfy the equation 4. In the process of re-training, the training data are re-selected randomly from the reserved labeled data. Random selection aims to avoid "teachers" being over-fitting and to improve the generalization of the model.

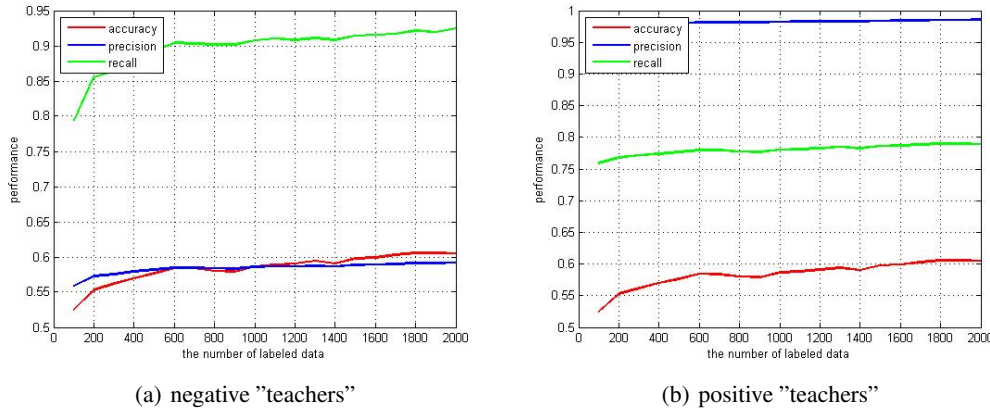


Figure 6. the average performances of negative and positive "teachers" trained in various data ranging from 100 to 2000.

It is fine that only a few labeled data are used to train these "teachers" because two weak classifiers could train a strong classifier according to Liu's work (Fuqiang Liu, 2016). Fig 6 shows the average performance of positive "teachers" and negative "teachers" trained by various numbers of labeled data. The number of training data ranges from 100 to 2000. Accuracy, precision and recall of these "teachers" are recorded. Based on Fig 6, it is concluded that the "teachers" could satisfy equation 4 when they are trained by a few labeled data.

6. Experiment

This section details the experiment about object recognition to test the performance of BPT multi-classifier. In the proceeding of training the multi-classifier, only a few labeled data (compared to the size of the whole data) are used. And besides the labeled data, the rest data in the data set are also used as unlabeled data. We use CIFAR-10 and CIFAR-100 (Krizhevsky, 2009) in the experiment and record the "accuracy" as the index of performance. In addition, we also compare the performance of the proposed method with those of previous and related methods.

In test, only 1000 labeled data are reserved to train these "teachers" in BPT. The rest of the training data (49000 totally) is used as unlabeled data. And the test set includes 10000 images. Table 2 shows the performances of the proposed method in CIFAR-10 and CIFAR-100. Besides, there are some other methods (Krizhevsky et al., 2012b; Kaiming He, 2015; Tsung-Han Chan, 2014; Mark D. McDonnell, 2015; Coates & Ng., 2011; Julien Mairal, 2014; Graham, 2015; Alexey Dosovitskiy & Brox, 2014; Krizhevsky et al., 2012a; Jasper Snoek, 2015; Ming Liang) compared with the proposed one in CIFAR-10. And we compare the proposed method with some other methods (Krizhevsky et al., 2012b; Rupesh Kumar Srivastava, 2015; Jost Tobias Springenberg, 2013; Min Lin; Shuo Yang & Tang, 2015; Tsung-Han Lin, 2014; Forest Agostinelli, 2015; ELU; Oren Rippel, 2015; Graham, 2015; Ming Liang) in CIFAR-100. It should be noted that the training data of other methods are 100% labeled, while the proposed method only use 2% labeled data and 98% unlabeled data in training set.

In this experiment, the performances of the proposed method could almost approach the baseline, AlexNet (Krizhevsky

Table 2. Performance of various methods in CIFAR-10 and CIFAR-100

Tested in CIFAR-10				
method	accuracy	method	accuracy	method
AlexNet (Krizhevsky et al., 2012b)	81.96%	proposed method	78.39%	DUFL (Alexey Dosovitskiy & Brox)
PCANet (Tsung-Han Chan, 2014)	78.67%	S-CNN (Mark D. McDonnell, 2015)	75.96%	CNN (Krizhevsky et al., 2012a)
K-means Net (Coates & Ng., 2011)	77.4%	Res-Net (Kaiming He, 2015)	93.57%	SBO (Jasper Snoek, 2015)
CK-Net (Julien Mairal, 2014)	82.18%	FMP (Graham, 2015)	96.53%	Rec-CNN (Ming Liang)
Tested in CIFAR-100				
method	accuracy	method	accuracy	method
AlexNet (Krizhevsky et al., 2012b)	54.2%	proposed method	50.77%	PMU (Jost Tobias Springenberg, 2015)
NIN (Min Lin)	64.32%	DRL (Shuo Yang & Tang, 2015)	64.77%	OMP (Tsung-Han Lin, 2014)
AF (Forest Agostinelli, 2015)	69.17%	ELUs (ELU)	75.72%	SP-CNN (Oren Rippel, 2015)
HighNet (Rupesh Kumar Srivastava, 2015)	67.76%	FMP (Graham, 2015)	73.61%	Rec-CNN (Ming Liang)

et al., 2012b). Especially in CIFAR-10, the proposed method performs even better than some supervised methods. In consideration of that the proposed method uses 2% labeled data and 98% unlabeled data, the performance of the proposed method is acceptable and reasonable. However, in CIFAR-100, the proposed method does not perform as good as it does in CIFAR-10. The main reason is that the ensemble strategy is not fit for the situation that the data includes too many classes, which is theoretically analyzed in section 4.3.

7. Conclusions

The paper researches in object recognition based on a few labeled data and amounts of unlabeled data. Different from previous works on object recognition based on semi-supervised methods or weak-supervised methods, the key idea of the proposed method is that a supervised classification model could be trained by amounts of unlabeled data with their estimated labels as well as the same model trained by all labeled data. To achieve this goal, we design Boost Picking Teaching for binary-classification. Kalal's P-N learning (Kalal et al., 2010; 2012) and Liu's Boost Picking (Fuqiang Liu, 2016) are the theoretical bases of BPT. Furthermore, how to expend binary-classification to multi-classification is discussed. This paper constructs an multi-classifier by synthesizing multiple binary-classifiers and analyzes the reason. In two object recognition data sets, BPT multi-classifier performs well even using just a few labeled data and amounts of unlabeled data. Especially when the number of class is less than dozens, the proposed model works effectively.

But there are two problems need further research. One is to design an ensemble method that is fit for many-class classification. Another one is improving the efficiency. BPT involves a lot of processes about re-training a classifier. The multi-classifier contains many BPT-trained binary-classifiers and every binary-classifier need a lot time to train well. The second one restrict the development of the proposed method seriously.

References

Fast and accurate deep network learning by exponential linear units (elus).

Alexey Dosovitskiy, Jost Tobias Springenberg, Martin Riedmiller and Brox, Thomas. Discriminative unsupervised feature learning with convolutional neural networks. NIPS, 2014.

Attenberg, Josh, Ipeirotis, Panagiotis G, and Provost, Foster J. Beat the machine: Challenging workers to find the unknown unknowns. *Human Computation*, 11:11, 2011.

Bertsekas, Dimitri P. Incremental gradient, subgradient, and proximal methods for convex optimization: A survey. *Optimization*, 2015.

Brooms, A. C. Stochastic approximation and recursive algorithms with applications, 2nd edn by h. j. kushner and g. g. yin. *Journal of the Royal Statistical Society*, 169(3):654–654, 2006.

Burton, Michele A, Brady, Erin, Brewer, Robin, Neylan, Callie, Bigham, Jeffrey P, and Hurst, Amy. Crowdsourcing subjective fashion advice using vizwiz: challenges and opportunities. In *Proceedings of the 14th international ACM SIGACCESS conference on Computers and accessibility*, pp. 135–142. ACM, 2012.

Chen, Xinlei, Shrivastava, A., and Gupta, A. Neil: Extracting visual knowledge from web data. In *Computer Vision, IEEE International Conference on*, pp. 1409–1416, 2013.

Coates, A. and Ng., A. Y. Selecting receptive fields in deep networks. pp. 2528C2536. NIPS, 2011.

Coates, Adam and Ng, Andrew Y. Learning feature representations with k-means. *Lecture Notes in Computer Science*, 7700:561–580, 2012.

Dalal, Navneet and Triggs, Bill. Histograms of oriented gradients for human detection. In *IEEE Conference on Computer Vision & Pattern Recognition*, pp. 886–893, 2005.

Fei-Fei, L. Imagenet: crowdsourcing, benchmarking & other cool things. In *CMU VASC Seminar*, 2010.

Felzenszwalb, Pedro F, Girshick, Ross B, David, Mc Allester, and Deva, Ramanan. Object detection with discriminatively trained part-based models. *Pattern Analysis and Machine Intelligence IEEE Transactions on*, 32(9):1627–1645, 2010.

Fergus, Rob, Weiss, Yair, and Torralba, Antonio. Semi-supervised learning in gigantic image collections. In *Advances in Neural Information Processing Systems 22: 23rd Annual Conference on Neural Information Processing Systems 2009. Proceedings of a meeting held 7-10 December 2009, Vancouver, British Columbia, Canada.*, 2009.

Forest Agostinelli, Matthew Hoffman, Peter Sadowski Pierre Baldi. Learning activation functions to improve deep neural networks. CVPR, 2015.

Fuqiang Liu, Fukun Bi, Yiding Yang Liang Chen. Boost picking: A novel method on converting supervised classification to semi-supervised classification. 2016.

Geman, Stuart, Bienenstock, Elie, and Doursat, René. Neural networks and the bias/variance dilemma. *Neural computation*, 4(1):1–58, 1992.

Graham, Benjamin. Fractional max-pooling. 2015.

Hosmer Jr, David W and Lemeshow, Stanley. *Applied logistic regression*. John Wiley & Sons, 2004.

James, Gareth M. Variance and bias for general loss functions. *Machine Learning*, 51(2):115–135, 2003.

Jasper Snoek, Oren Rippel, Kevin Swersky Ryan Kiros Nadathur Satish Narayanan Sundaram Md. Mostofa Ali Patwary Prabhat Ryan P. Adams. Scalable bayesian optimization using deep neural networks. ICML, 2015.

Jolliffe, I. T. Principal component analysis. *Springer Berlin*, 87(100):41–64, 2010.

Jost Tobias Springenberg, Martin Riedmiller. Improving deep neural networks with probabilistic maxout units. ICML, 2013.

- Julien Mairal, Piotr Koniusz, Zaid Harchaoui. Convolutional kernel networks. 2014.
- K. Zhou, J.C. Doyle and Glover, K. Robust and optimal control. 1996.
- Kaiming He, Xiangyu Zhang, Shaoqing Ren Jian Sun. Deep residual learning for image recognition. 2015.
- Kalal, Zdenek, Matas, Jiri, and Mikolajczyk, Krystian. Pn learning: Bootstrapping binary classifiers by structural constraints. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pp. 49–56. IEEE, 2010.
- Kalal, Zdenek, Mikolajczyk, Krystian, and Matas, Jiri. Tracking-learning-detection. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 34(7):1409–1422, 2012.
- Krizhevsky, Alex. Learning multiple layers of features from tiny images. 2009.
- Krizhevsky, Alex, Sutskever, Ilya, and Hinton, Geoffrey E. Imagenet classification with deep convolutional neural networks. pp. 1097–1105. NIPS, 2012a.
- Krizhevsky, Alex, Sutskever, Ilya, and Hinton, Geoffrey E. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pp. 1097–1105, 2012b.
- Li, Fei Fei, Rob, Fergus, and Pietro, Perona. One-shot learning of object categories. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 28(4):594–611, 2006.
- Liang, Xiaodan, Liu, Si, Wei, Yunchao, Liu, Luoqi, Lin, Liang, and Yan, Shuicheng. Computational baby learning. *Eprint Arxiv*, 2014.
- Lin, Tsung-Yi, Maire, Michael, Belongie, Serge, Hays, James, Perona, Pietro, Ramanan, Deva, Dollár, Piotr, and Zitnick, C Lawrence. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014*, pp. 740–755. Springer, 2014.
- Mark D. McDonnell, Tony Vladusich. Enhanced image classification with a fast-learning shallow convolutional neural network. 2015.
- Min Lin, Qiang Chen, Shuicheng Yan. Network in network.
- Ming Liang, Xiaolin Hu. Recurrent convolutional neural network for object recognition.
- Mohri, Mehryar, Rostamizadeh, Afshin, and Talwalkar, Ameet. *Foundations of machine learning*. MIT press, 2012.
- Oren Rippel, Jasper Snoek. Spectral representations for convolutional neural networks. NIPS, 2015.
- Prest, A., Leistner, C., Civera, J., and Schmid, C. Learning object class detectors from weakly annotated video. In *IEEE Conference on Computer Vision & Pattern Recognition*, pp. 3282–3289, 2012.
- Rosenberg, Chuck, Hebert, Martial, and Schneiderman, Henry. Semi-supervised self-training of object detection models. In *7th IEEE Workshop on Applications of Computer Vision / IEEE Workshop on Motion and Video Computing (WACV/MOTION 2005), 5-7 January 2005, Breckenridge, CO, USA*, pp. 29–36, 2005.
- Rupesh Kumar Srivastava, Klaus Greff, Jurgen Schmidhuber. Highway networks. 2015.
- Shalev-Shwartz, Shai. Online learning and online convex optimization. *Foundations and Trends in Machine Learning*, 4(2):107–194, 2012.
- Shuo Yang, Ping Luo, Chen Change Loy Kenneth W. Shum¹ and Tang, Xiaou. Deep representation learning with target coding. AAAI, 2015.
- Socher, Richard, Ganjoo, Milind, Sridhar, Hamsa, Bastani, Osbert, Manning, Christopher D., and Ng, Andrew Y. Zero-shot learning through cross-modal transfer. *Advances in Neural Information Processing Systems*, pp. 935–943, 2013.
- Sung, Kah Kay and Poggio, Tomaso. Example based learning for view-based human face detection. *Pattern Analysis and Machine Intelligence IEEE Transactions on*, 20(1):39–51, 1995.

- Suykens, Johan AK and Vandewalle, Joos. Least squares support vector machine classifiers. *Neural processing letters*, 9 (3):293–300, 1999.
- Tsung-Han Chan, Kui Jia, Shenghua Gao Jiwen Lu Zinan Zeng Yi Ma. Pcanet: A simple deep learning baseline for image classification? 2014.
- Tsung-Han Lin, H. T. Kung. Stable and efficient representation learning with nonnegativity constraints. ICML, 2014.